IN THE CLAIMS

Please amend the claims as follows.

1. (Currently Amended) A computer program product comprising:

semaphore means for stalling a first task until one of a predetermined set of events occurs, and

~~characterized in that the computer program product comprises specific~~ a plurality of message files associated with said first task for receiving data to be processed by the first task, the occurrence of one of said set of predetermined events causing a piece of data to be written in one of the associated message files;

wherein the message files are each given a priority level; and

wherein the first task is capable of processing the data in the message files in order of the priority of the message files.

2. (Original) A computer program product as claimed in claim 1, wherein said set of predetermined events includes hardware interruptions, a hardware interruption causing a selected interruption handler to write data into one of the message files associated with the first task, a blocking mechanism being used for temporarily blocking the interruptions during said data writing into the message file.

3. (Original) A computer program product as claimed in any one of claim 1 or 2, wherein said set of predetermined events includes at least an event causing a second task to write data into one of the associated message files.

4. (Cancelled).

5. (Currently Amended) A computer program product as claimed in claim [[4]] 1, wherein the message files with the highest priority level are allocated to interruption handlers so as to receive data from said interruption handlers.

6. (Original) A computer program product as claimed in any one of claims 1 to 5, wherein said message files comprise a data structure which is accessible by pointer manipulation.

7. (Original) A computer system for implementing a computer program as claimed in any one of claims 1 to 6, wherein said system comprises:

- a processor for executing a set of schedulable tasks including at least the first task,
- a scheduler for selecting one of said schedulable tasks for execution by said processor,
- at least an interrupt handler for handling interrupts,
- an interrupt server for performing a specific function in response to the occurrence of an interrupt.

8. (Original) An electronic apparatus comprising a computer system as claimed in claim 7.

9. (Currently Amended) A method of synchronizing a first task with respect to an occurrence of one of a predetermined set of events, the method comprising:

- a waiting step for making said first task wait for a specific semaphore until one of said predetermined set of events occurs, said first task being associated with a plurality of message files for receiving data in a message, said data reception being caused by the occurrence of one of said predetermined events, the message files each given a priority level,

- a blocking step for temporarily blocking interruptions during reception of said data,

- a reception step for receiving said data in at least one of the associated message files,

- a wakeup step for waking up said first task upon reception of the data in at least one of its associated files, and

- a reading and processing step performed by said first task for reading and processing the data received in [[a]] at least one of the message files, wherein processing the data comprises processing the data in the message files in order of the priority of the message files.

10. (Original) A method as claimed in claim 9, wherein said set of predetermined events includes hardware interrupts causing interrupt handlers to write data into one of the message files associated with the first task.

11. (New) The computer program product of Claim 1, wherein:

the message files comprise first-in, first-out data structures; and

an interrupt handler is capable of writing data to the first-in, first-out data structures.

12. (New) The computer program product of Claim 1, wherein the semaphore means comprise a counter capable of being incremented when a message is received in any of the message files and capable of being decremented when a message is read from any of the message files by the first task.

13. (New) The method of Claim 9, wherein:

the message files comprise first-in, first-out data structures; and

an interrupt handler is capable of writing data to the first-in, first-out data structures.

14. (New) A system, comprising:

a processor capable of executing at least one task including a first task and at least one interrupt handler;

a memory capable of storing a plurality of message files associated with the first task, the message files containing data to be processed by the first task, the message files each given a priority level;

a scheduler capable of controlling the at least one task; and

an interrupt server capable of providing an interrupt to one of the at least one interrupt handler for processing;

wherein the scheduler is capable of using a semaphore to stall the first task until at least one of a plurality of events occurs; and

wherein the occurrence of at least one of the events causes data to be written in at least one of the message files, the first task capable of processing the data in the message files in order of the priority of the message files.

15. (New) The system of Claim 14, wherein the plurality of events includes a hardware interrupt capable of causing one of the at least one interrupt handler to write data into one of the message files associated with the first task.

16. (New) The system of Claim 15, wherein a blocking mechanism is capable of temporarily blocking the hardware interrupts when data is being written into one of the message files.

17. (New) The system of Claim 14, wherein the plurality of events includes an event causing a second task to write data into one of the message files.

18. (New) The system of Claim 14, wherein one or more of the message files are capable of receiving data from the at least one interrupt handler and are assigned a highest priority level.

19. (New) The system of Claim 14, wherein the message files comprise first-in, first-out data structures.

20. (New) The system of Claim 19, wherein the at least one interrupt handler is capable of writing data to the first-in, first-out data structures.

21. (New) The system of Claim 14, wherein the semaphore comprises a counter capable of being incremented when a message is received in any of the message files and capable of being decremented when a message is read from any of the message files by the first task.